USING FORMAL CONCEPT ANALYSIS WITH A PUSH-BASED WEB DOCUMENT MANAGEMENT SYSTEM

BY

Timothy John Everts (BComp.)

A DISSERTATION SUBMITTED TO THE

School of Computing

IN PARTIAL FULFILMENT FOR THE DEGREE OF

Bachelor of Computing with Honours

University of Tasmania

(November, 2004)

Declaration

I declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution, and to the best of my knowledge, contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Timothy John Everts
November 3, 2004

Abstract

The significant increase in amount of information readily available on the World Wide Web (WWW) makes it difficult for users to locate the information they desire in a timely manner. Modern information gathering and retrieval methods focus on simplifying this task by enabling the user to retrieve only a small subset of information that is more relevant and manageable. However, often the majority of users will not find an immediate use for the information. Therefore, it is necessary to provide a method to store it effectively so it can be utilised as a future knowledge resource.

A commonly adopted approach is to classify the retrieved information based on its content. A technique that has been found to be suitable for this purpose is Multiple Classification Ripple Down Rules (MCRDR). MCRDR constructs a classification knowledge base over time using an incremental learning process. This incremental method of acquiring classification knowledge suits the nature of Web information because it is constantly evolving and being updated. However, despite this advantage, the classification knowledge of MCRDR is not often utilised for browsing the classified information. This is because MCRDR does not directly organise the knowledge in a way that is suitable for browsing. As a result, often an alternate structure is utilised for browsing the information which is usually based on a user's abstract understanding of the information domain.

This study investigated the feasibility of utilising the classification knowledge acquired through the use of MCRDR as a resource for browsing information retrieved from the WWW. A system was implemented that used the concept lattice-based browsing scheme of Formal Concept Analysis (FCA) to support the browsing of documents based on MCRDR classification knowledge. The feasibility of utilising classification knowledge as a resource for browsing documents was evaluated statistically. This was achieved by comparing the concept lattice-based browsing approach to a standard one that utilises abstract knowledge of a domain as a resource for browsing the same documents.

Acknowledgements

Firstly, I would like to thank my supervisor Dr. Byeong Ho Kang for his guidance, patience and support throughout the year. His ongoing assistance and encouragement were invaluable and helped me to persevere to this point. Also, the many discussions that we had together, whether online or via video conference, were often made all the more interesting with his use of humorous comments and explanations.

I would also like to thank the staff of the School of Computing for their assistance and support. Dr. Peter Vamplew and Jacky Hartnett for the significant amount of time and effort they allocated to the overall supervision of Honours; Christian McGee for his prompt technical support and assistance; and Andrew Spilling and Rick Smith for their technical support and help with the video conferencing equipment, especially at the most critical of times.

I also owe much thanks to my family. Their continued support, patience and encouragement was also greatly appreciated. I especially thank my youngest sister Olivia who always had a hug to share and was able to put a smile on my face, even after the most stressful of days.

I wish also to thank my fellow students of the School of Computing, many of whom have also gone through this experience. Sung Gun Hwang for his assistance and the invaluable provision of a tape recorder for recording my supervisor meetings; Sungsik Park for his provision of data and related programs used during my experimentation process; Paul Semmens for kindly proof reading sections of this thesis; and my fellow Honours students Leigh de la Motte, Rob Fearn, Chris MacLeavy, Barry Pearn (Masters) and Owen Richards, who were right there with me every step of the way.

Finally, I give thanks to the Lord Jesus Christ. Without His grace, guidance, and loving care, I would not have been able to persevere throughout this challenging experience.

Contents

List of Tables

List of Figures

Chapter 1
Introduction

The World Wide Web (WWW) has become the most popular information source for people today and is now the largest sharable and searchable repository of information (Park et al. 2003, p. 612; Kim and Compton 2004). InternetWorldStats (2004) estimates that between the years of 2000 and 2004, the world wide usage of the Internet grew by approximately 125 percent, with the biggest growth recorded in the Middle East region (227.8%). As the number of WWW users continues to increase, so also does the quantity of information available. As a result, much of this information is unstructured and decentralized in nature, meaning that it is becoming increasingly difficult for WWW users to locate the information they desire.

In order to tackle the information overload problem, two common approaches to searching for information on the WWW have evolved (B.H. Kang 2004, pers. comm., October 4). A user will either search for information knowing exactly what it is they are looking for, or they will visit various Web sites just to glance at the information available, so as to determine whether that information might be useful or relevant. The majority of users that adopt the first approach will normally utilise a search engine or other search service to locate the information they need (Kobayashi and Takeda 2000). However, this often results in a significant amount of irrelevant information being returned, and the majority of users will not even read beyond the first page of search results (Greenspan 2002). Users that adopt the second of the two approaches usually visit information rich Web sites, such as online newspapers or Web portals, just to see what information is currently available and whether it is of interest to them. While many of these types of sites provide a rich source of information for WWW users, the quantity of information available is often too large and the majority of it may not be immediately relevant to the user. This suggests that it would be ideal if such information could be archived for later reference because it may become more relevant to the user in the future. Therefore, the main inadequacies of both information retrieval approaches is the large quantity of information returned,

and thus the inability to produce information in a timely manner that is relevant to the individual WWW user.

In an attempt to help WWW users retrieve only information that they specify as relevant, Internet software companies in the late 1990's began developing applications that use what is now known as "push technology" (Buchwitz 1997). Software applications that utilise push technology automatically deliver (push) information to a user's desktop so that the effort is exerted by the publisher in sending the information to the user, rather than the user having to actively search (pull) for that information. As Chin (2003) notes, the advent of push technology was aimed at closing the gap between the time information is made available and the time a user retrieves it. Closing this gap would enable important information to be delivered to key-decision makers in real time. However, Chin also concludes that the use of push-based technology as a de facto method for Internet and Intranet information gathering and delivery is unlikely. This is mainly because the majority of users find push-based clients to be too obtrusive and the quantity of information produced is too overwhelming for the user to manage. Therefore, this implies that a more effective and advanced method of information delivery that uses the push technology is needed in order to make it truly useful and more widely accepted by users.

Since the use of push-based technologies for information gathering and delivery is not popular amongst WWW users, the majority of them continue to utilise information retrieval methods that are based on pull technology. One such example of this technology are traditional search engines (Lam and Ozsu 2002). However, this still does not resolve the problem of retrieving only the information that is relevant to the WWW user, especially since most search engines are configured to only return results from pages that have been registered with the search engine service. This means that the most current and therefore most relevant information, from newly updated and dynamically changing Web sites such as on-line newspapers, is often missed in the search results. Therefore, for users wishing to find newly updated information, a search engine cannot successfully fulfil the user's request (Park et al. 2003).

In order to overcome these sorts of problems with conventional, passive-based information delivery mechanisms, a more active mechanism was required. This stemmed the research and development of software applications that could deliver the most up to date information in a timely manner. Some of these could also be customised to manage or filter the amount of information being delivered. An example of such software that has become popular in recent times is Web Monitoring Systems (Glance et al. 2001; Boyapati et al. 2002; Dumais et al. 2003; Park et al. 2003; Chakravarthy et al. 2004). A Web Monitoring System (WMS) operates by checking a limited amount of predefined target Web pages, detecting changes in these pages automatically, and prompting users when these changes occur. The use of such systems appear to offer at least a partial solution to the problems of traditional information retrieval methods because the user has more control over the type and amount of information being delivered. It also ensures that the information being gathered is the latest and therefore, most relevant. However, the quantity of information being gathered can still be reasonably large. Subsequently, an effective method for storing and managing this information is also required, especially since many users may not find an immediate use for the retrieved information (B.H. Kang 2004, pers. comm., 26 March).

If information retrieved by the WMS can be stored effectively, it can also be utilised as a sharable knowledge resource in the future. Kim et al. (2004a) suggests that the most effective way to store information is through the efficient classification of retrieved documents. Traditionally, the dominant approach for classification is based on the content (text) of documents through trained classifiers using Machine Learning (ML) techniques because they achieve impressive levels of effectiveness (Senastiani 2002). However, although classification by ML has proved to be successful in some commercial or research applications (Mladenic 1999), it is not generally appropriate for classifying information from the WWW. This is because the classification knowledge created during the training process cannot usually cater for the dynamic nature of Web documents. New information is constantly being generated or it is being updated. For this reason, efficient classification of documents retrieved from the WWW requires a technique that can operate on a continual learning process. This enables incremental knowledge acquisition that suits the dynamic nature of Web document information (Kim et al. 2004a). One example of a

method that is known to successfully fulfil this task, is the knowledge acquisition and representation technique known as Multiple Classification Ripple Down Rules (Kang et al. 1995; Kang 1996).

The Multiple Classification Ripple Down Rules (MCRDR) knowledge acquisition and representation technique constructs a classification knowledge base incrementally over time through a process of differentiation by the expert. When the case-based reasoning system of MCRDR retrieves cases that are recognised by the expert as inappropriate, the expert simply identifies the important characteristics of the present case that distinguish it from existing cases. In this way, knowledge is acquired by the system and new rules are created accordingly. When applied to Web Monitoring Systems, this technique enables the MCRDR rule set to be developed and adapted to suit the dynamic nature of Web documents.

Despite the appropriateness of using MCRDR to classify the documents collected by Web Monitoring Systems, the technique has one major weakness. MCRDR does not directly organise the knowledge in a way that is suitable for browsing (Kim and Compton 2004, p. 204). As a result, the heuristic classification knowledge in an MCRDR knowledge base is not often utilised for browsing and searching the documents. Instead browsing and searching is facilitated through a structure based on some form of abstracted knowledge about the document domain that has been provided by the expert or user (Kim et al. 2004b).

Therefore, it is suggested that the classification knowledge acquired through the use of MCRDR may also provide a useful resource for browsing the retrieved documents. To this extent, the research undertaken in this study assessed the feasibility of utilising the heuristic classification knowledge of an MCRDR knowledge base as a resource for browsing documents in a specified domain. A system was developed and implemented that adopted the lattice-based browsing method of Formal Concept Analysis (Wille 1982; Wille and Ganter 1999) as a means of providing a browsing representation based on heuristic classification knowledge. Formal Concept Analysis has been shown by Kim (2003) and Cho (2003) to be quite successful for browsing documents in a specified domain.

A comparative statistical analysis was performed between the use of a traditional browsing structure (based on abstract knowledge of a domain), and the concept lattice structure of FCA (based on heuristic classification knowledge). This has been done to evaluate the feasibility of utilising heuristic classification knowledge for browsing Web documents.

The knowledge utilised in the investigation was that collected by a Web Monitoring System known as WebMon (Kim et al. 2004a). WebMon monitors and retrieves documents from the WWW and then classifies them to a storage folder structure using the MCRDR knowledge acquisition technique. The storage folder structure is also used by the system to facilitate browsing the retrieved documents. Note that the storage folder structure is a user devised structure based on abstract domain knowledge.

Chapter 2
Literature Review

2.1. WebMon Web Monitoring System

As mentioned at the end of Chapter 1, the source of heuristic classification knowledge utilised during this study was that collected by a Web Monitoring System called WebMon. For this reason an overview of the system is detailed in this section. A more detailed description can be found in Kim et al. (2004a).

The WebMon Web Monitoring System was developed by a number of researchers at the University of Tasmania, Australia, and was built as part of the Personalised Web Information Management System detailed in Park et al. (2003). WebMon focuses on the management of newly uploaded information for target Web sites on the WWW. To retrieve information from the WWW, it integrates the use of a particular model of push technology known as 'selective pull' (Buchwitz 1997). In the selective (automatic) pull paradigm, a user subscribes to certain types of information and then specifies when that information is to be delivered. In the WebMon system, this involves the user registering certain target Web sites for monitoring and then defining configurations for the monitoring of those sites. This includes parameters such as a monitoring schedule and also certain keywords or key phrases that are used by the system to identify retrieved pages that are of relevance to the user. This information is stored by WebMon in a user profile. Once the actual Web monitoring service begins, it is automated and will change only if the user requests it.

WebMon also utilises push technology to share the collected information with users (Park et al. 2003). It uses a combination of the 'distributed push/pull' and 'interactive push' models (Buchwitz 1997) to enable users to send and receive the information retrieved during Web monitoring. Users of WebMon can register recipients who should be emailed the newly retrieved information. This can be done either automatically or upon the recipient's request. The information can also be posted to a customisable Web portal that can be accessed by WWW users.

Since WebMon is based on push technology, it provides the WWW user with a more active method for information gathering and delivery. However, its use of push technology also means that the amount of information gathered is of a high quantity and this is usually somewhat overwhelming for users (Buchwitz 1997). Consequently, it is necessary that the system provides a method for archiving this information effectively so it can be utilised in the future. For this purpose, WebMon adopts the MCRDR knowledge acquisition technique to classify and store retrieved documents appropriately.

2.1.1. Multiple Classification Ripple Down Rules

Multiple Classification Ripple Down Rules (Kang et al. 1995; Kang 1996) is an incremental case-based Knowledge Acquisition (KA) methodology by which the expert can develop and maintain a case base without the help of knowledge engineers. The Multiple Classification Ripple Down Rules (MCRDR) method is derived from the Ripple Down Rules (RDR) method, a hybrid case-based and rule-based approach for knowledge acquisition and representation (Richards 2001). RDR was developed specifically from the experience gained in maintaining an early medical expert system known as GARVAN-ES1 (Compton and Jansen 1989; Kim et al. 2004a). The major success of the RDR approach has been its utilisation in a large medical expert system called PIERS. PIERS was built by experts without the support of a knowledge engineer (Edwards et al. 1993). However, the main disadvantage of the RDR method is that it only allows for single classifications of each case presented. As a result, the MCRDR method was developed, which allows for multiple independent classifications for each case, while still preserving the advantages and essential strategy of RDR.

Knowledge acquisition (KA) in MCRDR involves the incremental addition of cases and justifications (rules) in the circumstance where a case is misclassified by the MCRDR system in the retrieval process. This incremental approach to KA is centred on the idea that the knowledge an expert provides is essentially a justification for a conclusion in a particular context (Compton and Jansen 1989; Preston et al. 1996). When the case-based reasoning (CBR) system of MCRDR retrieves a case(s) that is incorrect, the expert is required to identify the important characteristics that

distinguish the incorrectly retrieved cases from the present case (Kang et al. 1997, p. 612). It is thought that experts will select more valid knowledge if asked to deal with the differences between cases (Kang et al. 1995). Thus, the expert's justification provides a basis for a new rule to be created. The new rule(s) is first validated against existing rules (cornerstone cases) and then automatically appended to the knowledge base.

The MCRDR knowledge acquisition technique is used by the WebMon Web Monitoring System for determining where documents retrieved during Web monitoring should be stored for archival and sharing purposes. The structure used by the system to store the information is a storage folder structure. It is comparable to a hierarchical tree arrangement of folders, much like that used in common operating system environments such as Microsoft Windows. Depending on the choice of the user, the entire storage folder structure can be defined up front or it can be defined incrementally as documents are collected. It is important to note that there are no predefined specifications that state the requirements for the specific folders contained in the storage folder structure. The structure is usually devised based on the user's knowledge or understanding of the monitored document domain. It should also be noted that if the user chooses to utilise the Web portal option to share the collected information with other users, this same storage folder structure is replicated on the Web portal site. It is provided as a means for browsing and searching for the documents.

Once the storage folder structure has been defined, newly updated Web documents retrieved during Web monitoring are classified into one or more target folders. Keywords are extracted from documents and form the conditions of rules in the MCRDR knowledge base. The rule conclusions are target folders in the storage folder structure. This means that keywords in a newly retrieved document can be utilised in inferencing the MCRDR knowledge base, in order to recommend a target storage folder for the document. In the circumstance when a document is misclassified as a result of the inference process, the user simply adds knowledge to the knowledge base that enables a correct classification to be made.

As an example of the inferencing process for a document, Figure 2.1 shows how a document with the case (keywords) of [a,b,c,d,e,f,g] is recommended to storage locations within the storage folder structure (SFS). The MCRDR KBS is drawn as an *n*-ary tree, with each node of the tree representing a rule which has a corresponding case. The inferencing process involves all rules attached to true parents being evaluated against the data. Thus the process begins by evaluating the root rule and then moving down level by level until either a leaf node is reached or none of the child nodes evaluate to true (Dazely and Kang 2003, p. 246). Since multiple pathways of refinement can be selected, multiple conclusions can be reached. This means that the last true rule on each pathway forms the conclusion for the case. Therefore, in the case presented in Figure 2.1, the inferencing process results in the recommendation of three storage folders for the current document, namely folders F_2, F_6, and F_5.

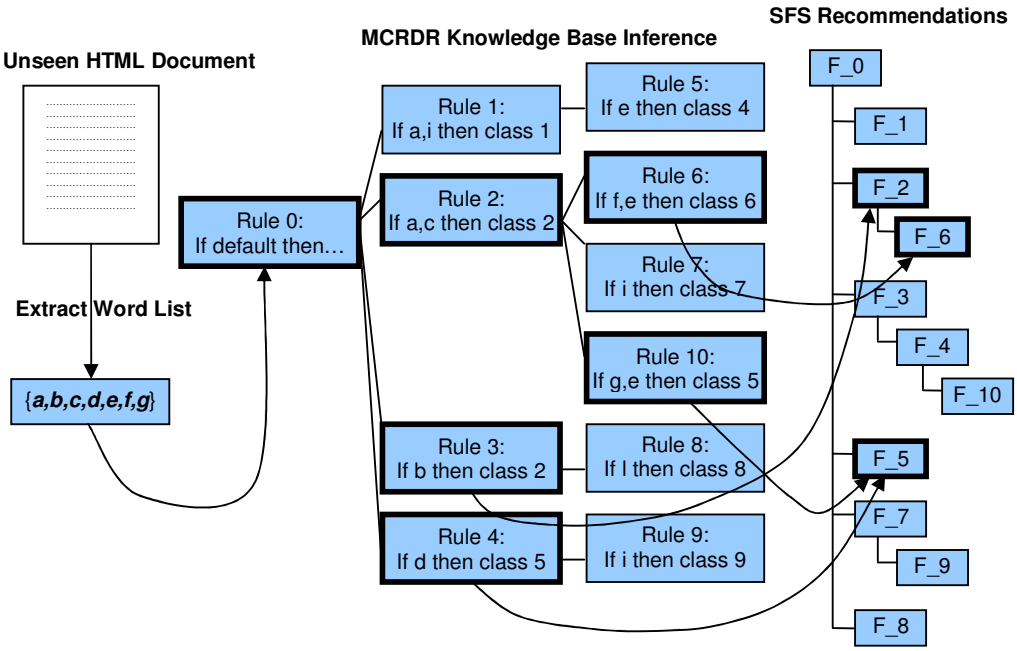

Figure 2.1 – Inference for a Web Document Classification; adapted from Kim et al. (2004a)

2.1.2. Knowledge Types

Analysis of the WebMon Web Monitoring System reveals that the user (or domain expert) is utilising the devised storage folder structure as a basis for defining a conclusion for document classifications. The common folder structure is used as a

mediating knowledge representation for the user, and it enables them to easily build a conceptual document classification model using folder manipulation (Kim et al. 2004b). In other words, the devised storage folder structure is an explicit representation of the user's knowledge of the current document domain. Evidently, two types of knowledge are actually being utilised in the classification process. One type of knowledge is being used to define the storage folder structure. Another type of knowledge is being used in the actual classification of documents to target folders. This point is more apparent when the user devises the storage folder structure. That structure is based upon their conceptual hierarchical understanding of the domain. However, when the user classifies a document to a folder in the storage structure, that classification is made based on the actual content of the document, namely keywords. These keywords may also be embedded in the conditions of the existing classification rules in the MCRDR knowledge base. Thus the distinction between the two knowledge types is clear.

The knowledge used in the creation of the storage folder structure is alluded to by Park et al. (2003) as being 'an abstract concept of special knowledge'. Kim et al. (2004a) implies that it can also be referred to as 'domain knowledge'. Therefore, this type of knowledge shall hereafter be referred to as being 'abstract domain knowledge' and the term is used frequently throughout the remainder of this thesis.

In regards to the second type of knowledge, it shall hereafter be referred to as being 'heuristic classification knowledge', since it is associated with the classification knowledge embedded in the rules of the MCRDR knowledge base. Kim et al. (2004a) also commonly refers to this type of knowledge as being 'classification knowledge', so the term 'heuristic classification knowledge' is considered to be appropriate.

Having discovered that there are two types of knowledge being utilised by WebMon for document classification, it is well worth noting that only the abstract domain knowledge is ever utilised for browsing the documents. This occurs when the WebMon user posts the retrieved information to a Web portal site and, consequently, the storage folder structure is utilised as a means for browsing the documents. Kim et al. (2004b) makes the comment this is commonly utilised for browsing representations by most Web portal sites because 'they do not sufficiently support

knowledge-base customisation'. Furthermore, it is also known that MCRDR does not directly organise the knowledge in a way that is suitable for browsing (Kim and Compton 2004, p. 204). Therefore, there are two potentially useful knowledge types which could be used as a basis for browsing documents, but only one of them is currently being utilised by the majority of Web portal sites. This means WWW users are being forced into searching for documents using a user-defined structure, which is based on abstract domain knowledge rather than on heuristic classification knowledge. It can be argued that the classification knowledge used to classify documents would be more appropriately used as a basis for browsing the documents, because it more accurately represents the actual content of each document. For this reason, the main suggestion of this research was that if the classification knowledge can be incorporated as the basis for a document browsing structure, it may also provide an extremely useful resource for browsing the documents in the domain. Therefore, it was proposed that the use of an alternate browsing method instead of the storage folder structure may enable classification knowledge to be utilised as a basis for browsing the documents classified by MCRDR. The approach suggested and adopted in this study was the lattice-based browsing scheme of Formal Concept Analysis, so therefore it is outlined in the section that follows.

2.2. Formal Concept Analysis

Kim (2003), Cho (2003) and various other researchers have shown that a quite successful method for browsing documents in a specified domain is the lattice-based browsing approach of Formal Concept Analysis (Wille 1982; Wille and Ganter 1999). Formal Concept Analysis (FCA) is a mathematical approach used for conceptual data analysis and knowledge processing. It has had numerous applications for data analysis and information retrieval in fields such as medicine, psychology, ecology, social science and political science (Kim and Compton 2004). The remainder of the description of FCA provided is based on the detailed descriptions of Wille (1982) and Tam (2004).

2.2.1. Overview

FCA 'formulates concepts in terms of objects and their properties or attributes, and provides a way of combining and organising individual concepts (of a given context) into [a] hierarchically ordered conceptual structure [known as a] … concept lattice

structure' (Rajapakse and Denham 2003, p. 29). Correia et al. (2003, p. 282) comments that concepts are necessary for expressing human knowledge and a formalisation of concepts acts as means of communicatively representing knowledge. FCA is based on a formal understanding of a concept as a unit of thought, comprising its extension and intension. The extension (extent) of a formal concept is formed by all objects to which the concept applies (a set of objects) and the intension (intent) consists of all attributes existing in those objects (a set of attributes).

2.2.2. Formal Context

The set of objects, set of attributes and the relations between an object and an attribute in a data set form the basic conceptual structure of FCA (known as a *formal context).* A formal context is defined as a triple (*G*, *M*, *I*) where *I* maps the relation between a set of objects *G*, and a set of attributes *M*. This is denoted formally as:

$$C = (G, M, I) \qquad (2.1)$$

where *C* represents the context. In order to express that a particular object *g* is in a relation *I* with a particular attribute *m*, the relation is given by:

$$(g, m) \in I \ \ or \ \ gIm \qquad (2.2)$$

and should be read as "the object *g* has the attribute *m*".

A formal context is often represented in a 2-dimensional matrix, called a *cross table* (Wille and Ganter 1999, p. 17). The rows of the cross table are headed by the object names and the columns are headed by the attribute names. Table 2.1 illustrates the context of dietary habits of four individuals, derived from an example presented by Tam (2004, p. 4). A cross (x) in the table indicates that an object *g* has an incidence relationship with an attribute *m.*

	Fish	Beef	Pork	Chicken
Fred	x			x
Jess	x	x	x	
Bob	x		x	x
Mel	x		x	x

Table 2.1 – Dietary habits of individuals; adapted from Tam (2004, p. 4)

2.2.3. Formal Concept

Once a formal context has been defined, all the *formal concepts* of the formal context can be derived. A formal concept is represented as a pair (A, B), where A is a subset of objects of the formal context and B is a subset of attributes of the formal context. In order for a pair (A, B) to be a formal concept, all attributes common to objects in A, the intent, and all objects common to attributes in B, the extent, must be the same. This duality relationship is formalised by:

1. Set of attributes common to the objects in A (*intent*)

$$A' = \{ m \in M \mid (g,m) \in I \text{ for all } g \in A \} \quad (2.3)$$

2. Set of objects common to the attributes in B (*extent*)

$$B' = \{ g \in G \mid (g,m) \in I \text{ for all } m \in B \} \quad (2.4)$$

2.2.4. Concept Lattice

The formal concepts of a formal context can be ordered and arranged hierarchically into a conceptual structure of FCA called a *concept lattice*. Ganter and Wille (1997, p. 6) comment that concept lattices are useful for unfolding given data, 'making their conceptual structure visible and accessible, in order to find patterns, regularities, exceptions etc.' Therefore, the concept lattice structure provides a means of revealing the implicit relationships between data that are not otherwise obvious.

The concept lattice is ordered by the smallest set of attributes (intent) between the concepts (Kim 2003) and thus maps an ordering from the most general to the most specific concept, top to bottom. To form the concept lattice, hierarchical subconcept-superconcept relations between all the formal concepts need to be found. This is formalised by $(A_1, B_1) \leq (A_2, B_2) : \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$ where (A_1, B_1) is called a *subconcept* of (A_2, B_2), and (A_2, B_2) is called a *superconcept* of (A_1, B_1). 'The relation $\leq$ is called the hierarchical order of the concepts' (Kim 2003, p. 55). In more simple terms, a superconcept is 'a set that has all of the members of another set and additional members' and a subconcept is 'a set that has fewer members than another set but all the members it has are contained in the other set' (Richards 1998, p. 153). Here the set is intended to refer to a set of attributes (intent).

When representing the concept lattice graphically, it is necessary to determine the predecessors and successors of each concept. Predecessor concepts for a particular concept in the lattice can be found by 'finding the largest subconcept of the intents for each concept' and successor concepts can be found by 'finding the smallest superconcept of the intents' (Richards 1998, p. 153). When the lattice is formed, the largest subconcept will be the top most concept in the lattice, called the *supremum*, and the smallest subconcept will be the bottom most concept, called the *infimum*.

Figure 2.2 shows the concept lattice for the dietary context presented in Table 2.1. Each node in the lattice represents a concept, with the object to attribute relation attached at each node respectively. Since the concept lattice is ordered by the smallest set of attributes (intent) between the concepts, the set {fish} is the set of attributes of the supremum.

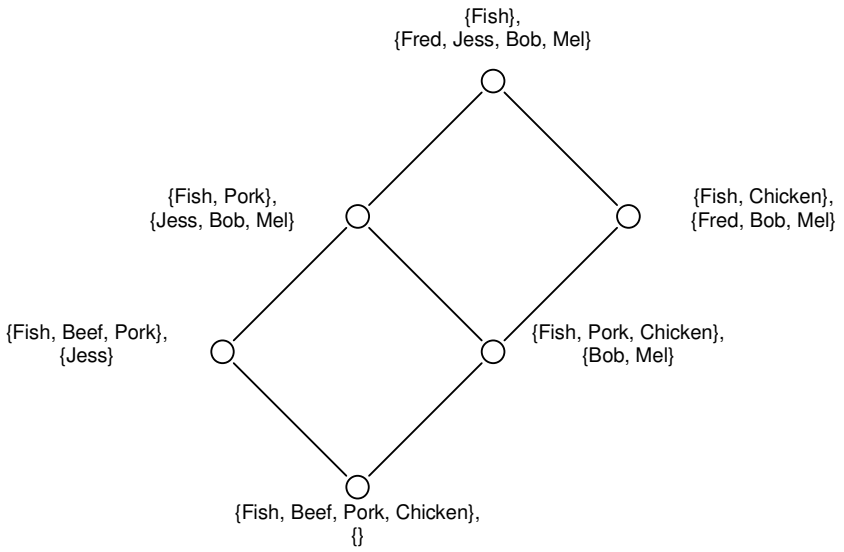

Figure 2.2 – Concept lattice for the context of Table 2.1; adapted from Tam (2004, p. 6)

2.3. Combining MCRDR with FCA for Browsing Documents

In this research, it was proposed that the lattice-based browsing approach of FCA can enable the MCRDR heuristic classification knowledge to be used as a resource for browsing the documents collected by the WebMon Web Monitoring System. For this reason, it is necessary to consider the appropriateness of combining the two techniques.

Various studies have shown that the lattice-based method of FCA can be utilised as an effective means for browsing documents in specialised domains. Kim (2003) developed a Document Management and Retrieval System (DMRS) for specialised domains on the WWW that utilised an incrementally built concept lattice as a means of browsing and retrieving documents. As part of her work, a user evaluation was performed on the browsing and retrieving of documents using the lattice structure. The evaluation concluded that users considered searching a specialised domain using lattice-based browsing to be more helpful than using Boolean queries and hierarchical browsing. Furthermore, users also found that the *ad hoc* evolvement of the lattice-based browsing structure provided good efficiency in retrieval performance.

Other research has revealed several other strengths of using lattice-based browsing. Kim and Compton (2004, p. 204) state that 'lattice browsing can help users to find both what they are looking for and also other useful documents beyond a narrow search'. They also comment that the hierarchically ordered relational structure of a concept lattice means that 'the user can navigate the parent concepts to search for more general documents or navigate the child concepts to get more specific documents' (pp. 219-220).

The lattice-based browsing approach has also been shown to be much more advantageous than a hierarchical approach to browsing documents, such as through the storage folder structure used by WebMon. Kim and Compton (2004, p. 216) write that:

> [in the lattice-based browsing approach,] if a relevant document is not found one can go back up another path rather than simply starting again. When one navigates down a hierarchy, one tries to pick the best child at each step. If the right document is not found, it is difficult to know what to do next, because one has already made the best guesses possible at each decision point.
>
> However, with a lattice, the ability to go back up via another pathway to the node opens up new decisions, which one has not previously considered.

In regards to utilising MCRDR classification knowledge in the lattice structure, research undertaken by Richards (1998) revealed that the rules of an RDR knowledge base can be utilised to generate an FCA concept lattice structure. Richards developed a system called MCRDR/FCA with the aim of reusing knowledge for activities that are not well supported by the performance knowledge captured using MCRDR. FCA was also used in the system as a way of uncovering the higher-level abstractions in the RDR knowledge base structure (KBS), as well as uncovering the structure between all concepts in the KBS that support the reflective modes. The RDR knowledge reused in the system was that from the medical expert system GARVAN-ES1 (Compton and Jansen 1989). Evaluation of the work undertaken consisted of consulting experts about the discovered concepts and carrying out a student survey comparing the number of different rule representations. The evaluation results strongly suggested that the integrated MCRDR-FCA system provided the types of knowledge acquisition and reuse required for the reflective activities.

Therefore, since the lattice-based browsing approach of FCA has been proven as an effective way of browsing documents in a domain, and since it has been shown that Ripple Down Rules (RDR) can be successfully combined with FCA to generate a lattice structure, the following conclusion was drawn for the study undertaken. The lattice-based browsing method of FCA may be used as a means for defining an effective document browsing structure that is based on MCRDR heuristic classification knowledge. The feasibility of this could be tested by utilising the structure to browse the documents collected by the WebMon Web Monitoring system and comparing this to browsing the same documents using the system's storage folder structure.

Chapter 3
Methodology

3.1. Overview

Having synthesised the various literature in Chapter 2, this chapter outlines the specific work undertaken to evaluate the feasibility of using MCRDR heuristic classification knowledge as a resource for browsing documents. It details a system that was implemented to facilitate the browsing using an FCA concept lattice and also outlines the strategy used for the evaluation.

It was proposed that the lattice-based browsing approach of FCA may facilitate effective browsing of documents of a specific domain. This proposition was based on utilising the heuristic classification knowledge of a MCRDR knowledge base for browsing. In order to test this theory, a source of MCRDR heuristic classification knowledge was required, together with a system that implemented the FCA lattice-based browsing approach by utilising that knowledge. Furthermore, to actually determine the feasibility of utilising heuristic classification knowledge for browsing documents of a specific domain, an evaluation had to be performed. This presented two basic options for an evaluation. One option was to perform a quantitative user evaluation. This would involve assessing the performance of browsing for documents in a concept lattice structure based on heuristic classification knowledge. The other option was to statistically analyse and compare the lattice structure with another browsing structure, which alternatively uses abstract domain knowledge as a basis for browsing the same documents. In analytically comparing the two different browsing structures, it is still possible to evaluate whether one is just as feasible than the other for browsing. This is because the actions a user can take when browsing is determined by the overall browsing structure itself anyway. Therefore, in this study the latter approach to evaluation was utilised, not only because it was a feasible approach, but also because there is a lot of difficulty involved with setting up and performing an effective user evaluation.

Thus, the following was the proposal for evaluating the feasibility of utilising heuristic classification knowledge for browsing documents of a specific domain. Since the WebMon Web Monitoring System (see section 2.1) utilises the MCRDR technique to classify retrieved documents to a storage folder structure (which is based on abstract domain knowledge), it was seen as an ideal candidate to be used for the purposes of the research investigations. Not only did WebMon provide the source of MCRDR classification knowledge that can be utilised for browsing documents in the concept lattice structure, but it also provided a browsing structure based on abstract domain knowledge that could be utilised as part of the evaluation process. Therefore, a system was implemented that utilised the MCRDR heuristic classification knowledge to generate a concept lattice structure for browsing the documents. Then a comparative statistical analysis was performed between browsing the generated concept lattice structure as opposed to browsing the storage folder structure. This was done in order to assess the feasibility of utilising heuristic classification knowledge as a resource for browsing documents.

3.2. MCRDR Heuristic Classification Knowledge Source

The MCRDR heuristic classification knowledge utilised in this research study was collected over a period of time during a project undertaken at the University of Tasmania in Hobart, Australia. During the project, WebMon monitored various target Web sites in the domain of eHealth and retrieved newly updated documents from these Web sites. These documents were classified to a user devised storage folder structure using recommendations provided from inferencing the rules in WebMon's MCRDR knowledge base. The MCRDR knowledge base was constructed incrementally as individual documents were classified. Incorrect recommendations made by the MCRDR inference engine were corrected by the user, and thus new knowledge was added to the knowledge base and utilised for future classifications of documents.

Table 3.1 summarises the data created as a result of the Web monitoring project. In total, 7 sites were monitored by WebMon and 7588 documents were retrieved from those sites. Of those 7588 documents, 4598 were classified to the storage folder structure which contained 119 folders. During the classification process, 172 rules

were created and a total of 285 unique rule conditions (keywords) were contained in those rules.

Web Monitoring	
Total eHealth Sites Monitored	7
Total Articles Retrieved	7588
Total Articles Classified	4598
Classification Knowledge	
Total Rules Used	172
Total Rule Conditions	285
Storage Folder Structure	
Total Folders	119

Table 3.1 – Summary of Web Monitoring Project

To store the data utilised or collected during the Web monitoring project, the WebMon system used a MySQL database. The data stored in the database included data about the Web sites monitored by WebMon, the documents collected during monitoring, the storage folder structure devised for storing and browsing documents, the MCRDR knowledge base containing the heuristic classification knowledge, and various other core system configurations. WebMon interfaced with this MySQL database in order to operate. The database was also utilised as a source for presenting the information to WWW users using a PHP-based Web portal site called iWeb. WWW users utilised the storage folder structure as a means for searching and browsing for the documents. The iWeb site divided the complete storage folder structure into various sub-domains of eHealth, based on the individual folders at the second level of the storage folder structure. These sub-domains included 'Diseases', 'Demographic Groups', 'Drug Information' and 'Health and Wellness'. Dividing the complete storage folder structure into smaller parts simplified browsing for information, especially since the entire storage folder structure was quite large and the quantity of information was significant.

Figure 3.1 shows the interface on the iWeb Web Portal Site for browsing documents in the 'Diseases' sub-domain. The storage folder structure is presented to the user on the left side of the interface. It is important to note that since the browsing of documents on the site is facilitated through traversing the storage folder structure, only the abstract domain knowledge represented in the structure is being utilised as the resource for browsing the documents. This means that although heuristic

classification knowledge had been used to store each document to an appropriate target folder, the knowledge is not currently utilised by the system as a means for browsing the documents.

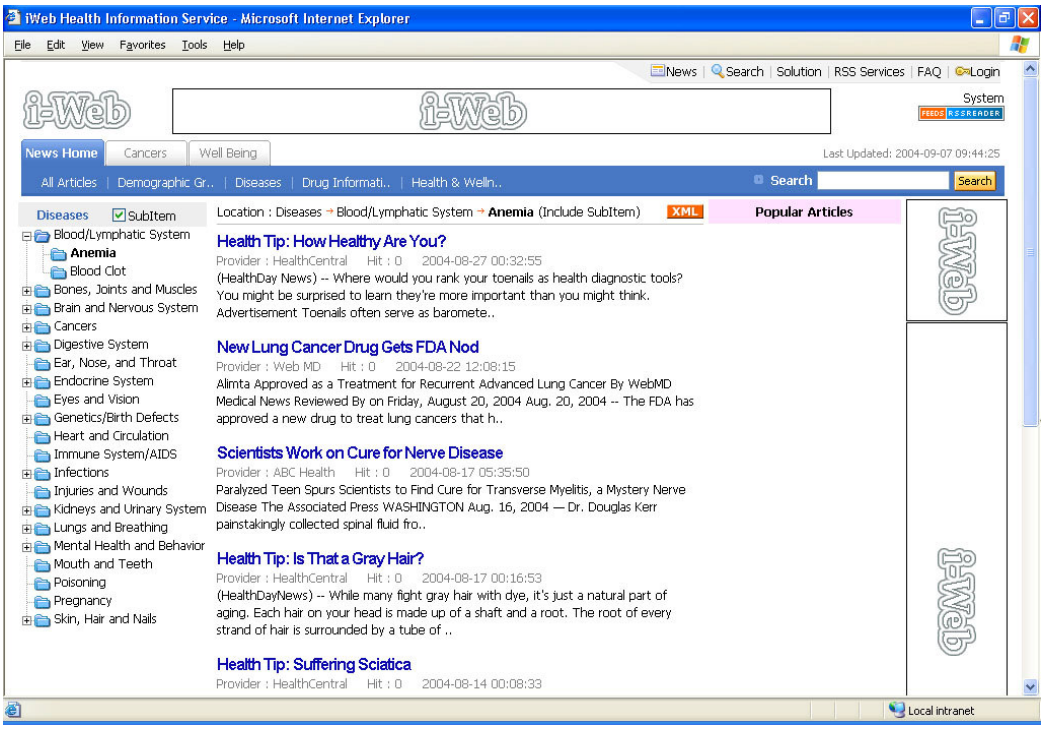

Figure 3.1 – iWeb Web Portal Site (Diseases sub-domain section)

3.3. Research Implementation

In order to utilise the MCRDR heuristic classification knowledge as a basis for browsing the documents collected during the Web monitoring project, it was necessary to develop a system that implemented an alternate browsing representation. Subsequently, a system was developed as part of this research which utilised the MCRDR heuristic classification knowledge to generate a FCA concept lattice for browsing the documents.

3.3.1. System Overview

To be able to utilise the heuristic classification knowledge in WebMon's MCRDR knowledge base it was necessary to implement a system that could directly access that information. Since the heuristic classification knowledge and all other data associated with the WebMon system was stored in a MySQL database, it was appropriate to implement a system that could directly interface with the database

content. For this reason it was considered appropriate to develop a Web-based system, since many Web-based programming languages provide functions that enable direct manipulation of data in a MySQL database. As a result, a Web-based prototype system named iWeb FCA was developed to fulfil the purposes of this research. The iWeb FCA system generates a FCA concept lattice based on the MCRDR heuristic classification knowledge to provide an alternate browsing structure for the documents collected and classified by WebMon. In addition, the system is also capable of utilising the abstract domain knowledge embedded in the storage folder structure as a resource for generating a concept lattice. The system can be configured to generate a concept lattice using either one of the knowledge sources as a resource or it can be configured to utilise both resources at once for lattice generation. This additional functionality is included in the system because it was discovered that often the names of the folders in the storage folder structure (abstract domain knowledge) are also utilised as keyword conditions in the classification rules (heuristic classification knowledge). This means some of the abstract domain knowledge can also be considered as being heuristic classification knowledge, especially since often the user creating the storage folder structure does not recognisably distinguish between these two knowledge types (B.H. Kang 2004, pers. comm., October 14).

In using the system to generate a concept lattice, it is important to note that documents are considered to constitute the objects used in FCA and the rule keywords (classification knowledge) or folder names (abstract domain knowledge) are considered to constitute the attributes. However, this approach does not strictly comply with the original formulation of FCA in which an object was implicitly assumed to have some sort of unity or identity so that the attributes applied to the whole object (e.g. a car has four wheels). As Kim (2003, p. 73) states, ‘clearly documents do not have the sort of unity where attributes will necessarily apply to the whole document’. However, in order to use FCA in the iWeb FCA system, the following assumptions are made. Documents correspond to objects and the rule condition keywords used to classify a document or the names of the folders in which the document is stored constitute the attribute set. A similar approach has been shown by Kim (2003) to be quite feasible.

3.3.2. Interface Design

A screen capture of the main menu of iWeb FCA is shown in Figure 3.2. In designing the iWeb FCA system, the major focus was on the functionality of the system rather than the actual layout or design of the system interface. This is because the system was built with the intention of simply providing the functionality to generate a concept lattice based on MCRDR classification knowledge. Therefore, this meant that it was unnecessary to consider usability as a priority in the design of the system.

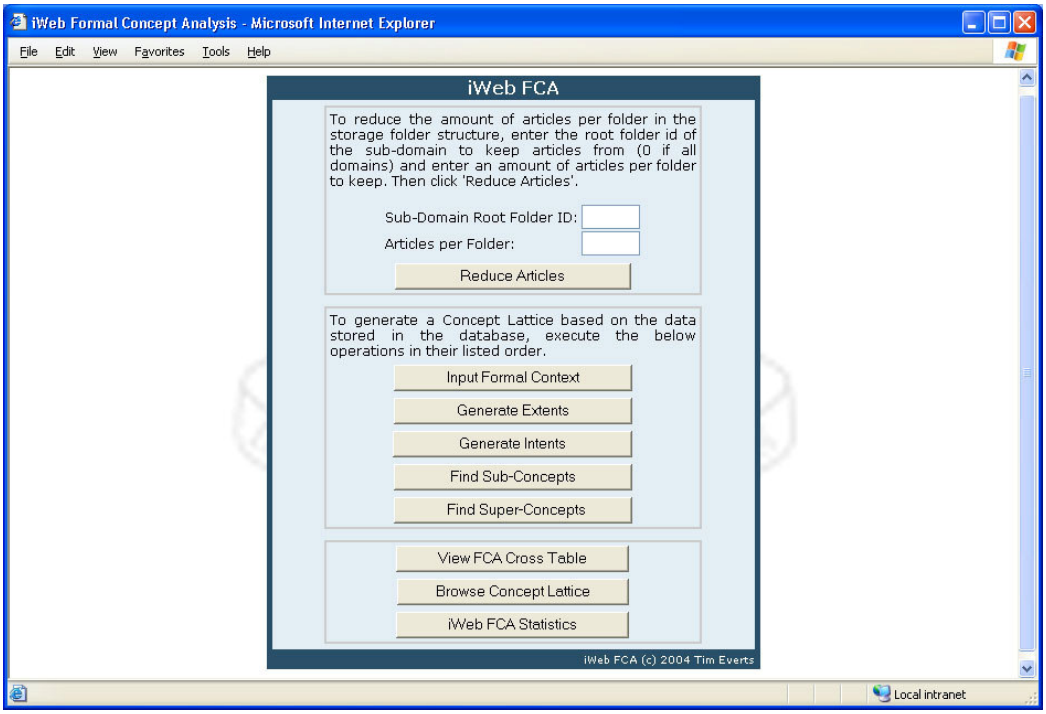

Figure 3.2 – iWeb FCA Main Menu

3.3.3. System Functionality

From the iWeb FCA main menu shown in Figure 3.2, there are four main operations that can be performed. Briefly, these operations are as follows:

1. Reduce the amount of documents in the specific domain (or sub-domain) to an amount from which a complete generation of a concept lattice is possible in a reasonable amount of time;
2. Generate a complete concept lattice based on the classification knowledge or abstract domain knowledge contained in WebMon's MySQL database;

3. View the generated concept lattice using a Web-based interface that utilises hyperlinks and URLs, and
4. View statistics on the structure of the generated FCA lattice and on the associated storage folder structure to which the documents were originally classified into and which is also used for browsing the documents on the iWeb Web portal site.

The functions are outlined in more detail in the sections that follow.

3.3.3.1. Reducing the Amount of Documents in the Domain

In order to evaluate the feasibility of utilising heuristic classification knowledge for browsing documents using an FCA lattice structure, it was only necessary to generate a single complete lattice for any formal context and gather statistical results about that generated lattice structure. However, the lack of available system resources and the significant quantity of documents for a single domain posed a problem for lattice generation. It was too time consuming to generate a complete concept lattice using the full set of documents. For this reason, iWeb FCA included a function that reduced the number of documents stored in all folders in the storage folder structure to contain, at a maximum, a specified amount. At a minimum, a folder could contain zero documents. Note the fact that the actual number of folders is not reduced meaning that all of the heuristic classification knowledge is still utilised to generate the complete concept lattice. This is because the MCRDR rules apply to particular folders in the storage folder structure, and not particular documents. In other words, the conclusions of the MCRDR rules are folders.

3.3.3.2. Generating a Complete Lattice

The main function of iWeb FCA is to generate a concept lattice based on MCRDR heuristic classification knowledge or abstract domain knowledge. However, before a concept lattice can be generated, it is necessary to have a formal context from which all formal concepts can be derived. For this reason, the formal context is stored by iWeb FCA in a MySQL database so the data of the formal context can be manipulated to generate the concept lattice.

To input the formal context, all objects and attributes that make up the context are determined. This is achieved by configuring the system to retrieve all the documents and classification or abstract domain knowledge for a given eHealth sub-domain from the WebMon MySQL database. Firstly, iWebFCA extracts all the documents for the particular domain and stores them as objects in its database. Then the attributes of those documents are extracted. Depending on the current system configuration, this means that all the rule keywords used to classify each document are extracted as the attributes of a document. The process used to extract the rule keywords is similar to that used by Richards and Compton (1997). The MCRDR knowledge base is converted to a flat structure by sequentially traversing the knowledge base for each rule and picking up the conditions from the parent rule until the top node with the default rule is reached. A similar method is used when the system is configured to also include the folder names of the storage folder structure as attributes of the document. In this case, the storage folder structure is converted to a flat structure by sequentially traversing the structure for each folder and picking up the names of parent folders until the root folder is reached. Once the formal context has been stored by iWeb FCA, it can then be used as a basis for computing the formal concepts and building a concept lattice.

Kim (2003) indicates that most algorithms proposed in the literature for computing formal concepts and building a concept lattice are either batch or incremental algorithms. She (pp. 56-57) defines batch algorithms as being algorithms that 'build formal concepts and a concept lattice from the whole context in a bottom-up approach (from the maximal extent or intent to the minimal one) or a top down approach (from the minimal extent or intent to the maximal one)'. On the other hand, 'incremental algorithms gradually reformulate the concept lattice starting from a single object with its attribute set'.

Incremental algorithms have been commonly utilised in the past for computing formal concepts and building a concept lattice because they have been known to perform with a higher level of efficiency than batch algorithms (Godin et al. 1995). However, incremental algorithms focus more on adding a new object into the lattice (Kim and Compton 2004). Since it would only be necessary to generate a single complete lattice for each formal context in this research, an incremental approach to

lattice generation was not required. Instead, a bottom-up batch process for generating the concept lattice was utilised. iWeb FCA builds the formal concepts and a concept lattice from a whole context, from the maximal extent to the minimal one. The generated concept lattice is then stored in iWeb FCA's database and represented via a Web-based interface for browsing.

The batch process utilised to build the formal concepts and the concept lattice is an implementation of the general methodology of FCA for formulating concepts and building the concept lattice. The algorithm used in iWeb FCA was based upon the explanations of FCA provided by Richards (1998), Kim and Compton (2000), and Kim (2003) (see also section 2.2). In detailing the procedure, *C* represents the formal context stored in iWeb FCA's database, *D* represents the set of objects (documents) in *C*, and *M* represents the set of attributes (rule keywords or folder names) in *C*. The procedure implemented is detailed in Figure 3.3.

Step 1:

Formulate an extent containing the set of objects *G* representing the largest concept of *C*. Then perform step 2 for each attribute *m* in the set *M*.

Step 2:

a) Find the set of objects X that contains the attribute *m*.

b) Check whether any previously formulated extent is equivalent to X.

c) If an equivalent extent of X does not exist, then add the set X as an extent of the attribute *m*.

d) Determine the intersection of X with all extents calculated in previous steps. If the intersection set does not exist, then add the intersection set as an extent of attribute *m*.

Step 3:

For each formulated extent, determine its intent: $Y \leftarrow \{ m \in M \mid (g,m) \in I \text{ for all } g \in X \}$

Step 4:

Construct the concept lattice by finding all the hierarchical subconcept-superconcept relations between all the formal concepts of *C* that were computed in steps 1 to 3.

Figure 3.3 – Procedure for Generating a Concept Lattice in iWeb FCA

3.3.3.3. Browsing the Concept Lattice

Once the concept lattice structure has been generated using the process outlined in section 3.3.3.2, the constructed lattice can also be viewed and browsed. However, it should be highlighted at this point that the original intention of implementing iWeb FCA just for generating the concept lattice and not for browsing it. The initial intention was to integrate the browsing of the concept lattice into the iWeb Web portal site, so as to provide an alternate browsing representation on that site instead of the storage folder structure. However, due to the limited amount of time available during the research, this integration could not completed. As a result, a prototype browsing interface was developed and incorporated into the actual iWeb FCA system. Developing the interface beyond the prototype stage and integrating it into the iWeb Web portal site may form part of future work.

In designing the lattice-based browsing interface for iWeb FCA, it was necessary to consider the various different ways a concept lattice can be represented for browsing. The most attractive method is to represent the entire lattice in the browsing interface because it 'can help the user in understanding the structure of the lattice and in seeing the naturally arising clusters and hierarchies' (Godin et al. 1989, p. 34). However, viewing the entire lattice on a single interface is often not practical because, as Godin et al. (1989, p. 34) write, 'the structure is usually too large for the limited display resolution and surface'. Consequently, a popular approach has been to decompose the lattice into smaller parts (Wille 1989) or show just a small portion of the concept lattice at once using a fisheye type view (Furnas 1986). Another approach has been to limit the number of concepts to a manageable size that can be viewed in a lattice by having the end-user specify certain concepts to display (Richards and Compton 1997). However, most of these proposals do not specifically focus on Web-based browsing of concept lattices. Since iWeb FCA is Web-based, it was important to select a representation that suited the nature of Web-based browsing and Web interfaces.

In recent work undertaken by Kim and Compton (2000; 2001), a Document Management and Retrieval System was developed that utilised a Web-based representation of the concept lattice. The concept lattice was represented using

hyperlinks and URLs as opposed to using a graphical lattice representation. This decision was made because it was 'anticipated that most Web users would have little familiarity with lattice displays' (Kim and Compton 2004, p. 235). The hyperlink technique was also well accepted by users for browsing the lattice and it was considered to be 'a fairly natural simplification for a lattice display' that didn't lose any of the advantages of FCA. Therefore, such a method was considered to be an appropriate one to be utilised in the context of the iWeb FCA system. As a result, a similar implementation is adopted for browsing the concept lattice generated by the system.

A sample of the concept lattice browsing interface used in iWeb FCA is shown in Figure 3.4. As in the approach of Kim and Compton (2000), the lattice display is simplified by showing only direct neighbour nodes of the current node using hyperlinks. Each lattice node represents a concept comprised of a pair (X,Y), where X is the extent (a set of documents) and Y is the intent (a set of classification rule keywords) of the concept. The intents of each concept are used for indexing the terms of the browsing structure.

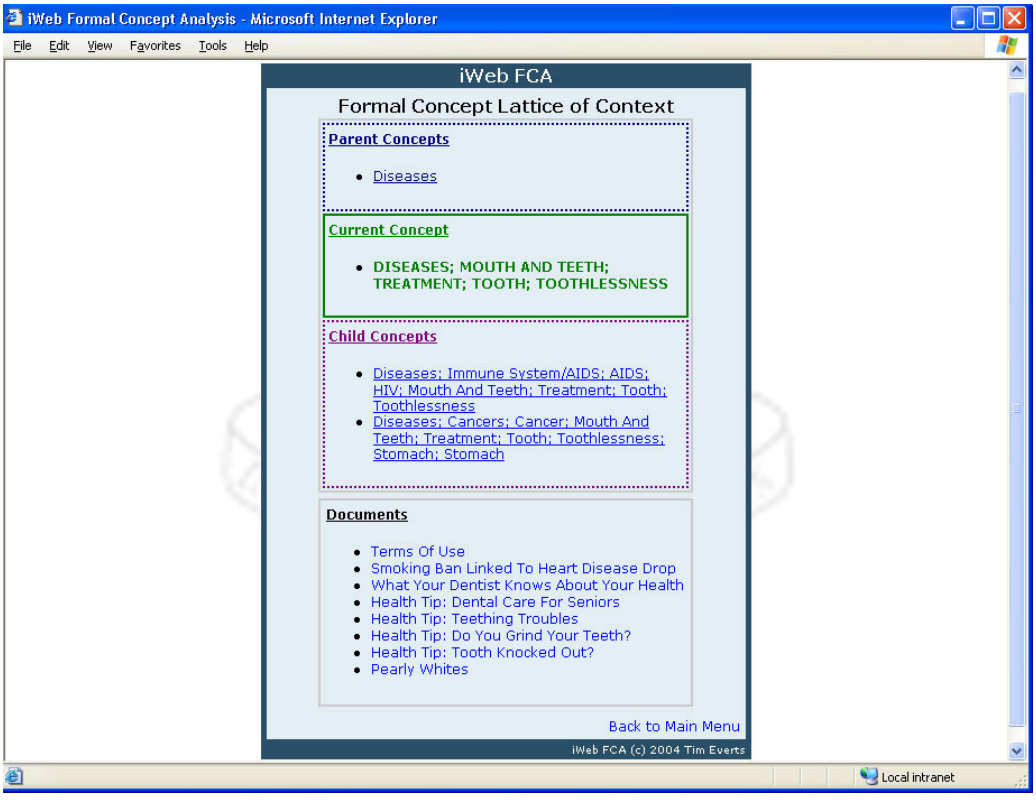

Figure 3.4 – iWeb FCA Concept Lattice Browsing Interface

The concept lattice browsing interface in iWeb FCA is divided into four distinctly recognisable sections (see Figure 3.4). The current lattice node is displayed in green in a section labelled 'Current Concept', while parent nodes and child nodes are listed as hypertext links in sections labelled 'Parent Concepts' and 'Child Concepts' respectively. The set of documents associated with the current node are listed as hypertext links in a section labelled 'Documents'. Note that the label and the border of each section is represented by a different colour, with the 'Current Concept' section using green, the 'Parent Concepts' section using navy, and the 'Child Concepts' section using violet. This design is based on the ideas of Kim and Compton (2000) whereby using different colours can facilitate a user's understanding of parents and child concepts.

The actual browsing of the lattice begins from the root node (concept) and the relationships of concepts can be explored by traversing from vertex to vertex by clicking on a child or parent node hypertext link. Each time a new node is selected, the interface is updated to show the parent and child nodes of the current node. The list of documents associated with the current node is also refreshed. Documents at a node can be viewed by clicking the appropriate hypertext link and the document will be displayed in a new Web browser window.

3.3.3.4. Gathering Statistics on Browsing Structures

The final of the four primary functions of iWeb FCA is included to assist with performing the evaluation of utilising heuristic classification knowledge for browsing. Using several built in formulas, iWeb FCA gathers statistics on the composition of the storage folder structure and concept lattice structure associated with a particular formal context. These statistics were used in the research evaluation to assess the feasibility of utilising heuristic classification knowledge for browsing. The type of statistics gathered include an analysis of the physical structures of the storage folder structure and concept lattice, an analysis of the distribution of documents in the two structures, and an analysis of browsing for documents based on the composition of each structure. Examples of these statistics are detailed in Chapter 4 of this thesis.

3.4. Evaluation Strategy

As stated in section 3.1, the approach adopted for assessing the feasibility of utilising MCRDR heuristic classification knowledge for browsing documents of a domain would involve performing a statistical comparative analysis between the generated concept lattice structure and the storage folder structure. To fulfil this, a sub-domain of the eHealth domain was first selected to be utilised as the source of data for generating the concept lattice. The reason why only a sub-domain was selected is because the limited system resources available meant it would take a significant amount of time to generate a single complete concept lattice for the entire eHealth domain. Also, since the storage folder structure could be distinctly divided into the various sub-domains of eHealth (as is done on the iWeb Web portal site), it was much simpler to just deal with a small portion of the overall structure for the purpose of analysing it. Consequently, the sub-domain of 'Diseases' was selected for the purpose of the analysis. It contained the most information out of all the sub-domains and also had the largest storage folder structure. To enable a concept lattice to be generated from the Diseases sub-domain data, iWeb FCA was used to reduce the number of documents in any folder to be no more than 32. This figure was chosen through a trial and error approach based on the amount of time it took to generate a concept lattice with the available system resources. It resulted in a total number of 1063 classified documents making up the reduced data set.

Having reduced the source domain data to a manageable amount for lattice generation, iWeb FCA was used to generate two different types of concept lattices. The first concept lattice was generated based on the MCRDR heuristic classification knowledge, and the second concept lattice was generated based on a combination of MCRDR heuristic classification knowledge (rule keywords) and abstract domain knowledge (folder names). As is discussed in section 3.3.1, many of the folder names used in abstract domain knowledge also occur as keywords in the heuristic classification knowledge. For this reason, it may also be potentially useful to browse documents using a combination of the two knowledge types, especially because often a user does not make a clear distinction between the two knowledge types (B.H. Kang 2004, pers. comm., October 14). Therefore, browsing a concept lattice based

on this combination of knowledge types was also assessed as part of the evaluation undertaken.

The final step of the evaluation procedure was to gather and record statistics on the different browsing structures. This was done in order to assess the feasibility of utilising heuristic classification knowledge for browsing documents. Three main forms of analysis were performed. Firstly, the physical composition of the different browsing structures was analysed as a means of assessing the implications that each would have on browsing for documents. Secondly, the distribution of documents in the browsing structures was compared to determine whether utilising heuristic classification knowledge as a resource for browsing enhances a user's ability to locate a particular document. Finally, an analysis was performed on how the structures would actually be browsed. This was achieved by programmatically simulating the browsing process and recording information about each level that would be traversed in each browsing structure. The results and discussion of the analytical evaluation are presented in Chapter 4.

Chapter 4
Results and Discussion

4.1. Overview

This chapter presents the results and discussion of the statistical evaluation performed in this research. Three main types of analysis were undertaken and in each analysis three different document browsing structures were compared. An analysis was performed on the physical composition of each structure, the distribution of documents in each structure, and a programmatic simulation of browsing each structure.

4.2. Analysis of Physical Browsing Structures

The first statistical analysis undertaken compared the physical composition of the storage folder structure (based on abstract domain knowledge) with the physical composition of a concept lattice structure based on the MCRDR heuristic classification knowledge. The aim of this analysis was to assess the implications that the different physical structures would have on browsing for documents.

4.2.1. Comparison of Storage Folder Structure and Lattice Generated using Heuristic Classification Knowledge

Table 4.1 shows the main statistics gathered from analysing the physical composition of the storage folder structure (SFS). Table 4.2 shows the statistics gathered from analysing the physical composition of a concept lattice which was generated based on the MCRDR heuristic classification knowledge (HCK lattice).

Total Number of Folders	80
Folders with Documents	56
Folders without Documents	24
Average Sub-Folders per Folder (without leaf folders)	6.08
Total Rules Utilised	78
Total Rule Keywords	109

Table 4.1 – Summary of Storage Folder Structure

Before comparing the physical composition of the SFS and HCK lattice, it is interesting to note that the SFS contains 80 folders in total but only 56 of them contain classified documents. This is most likely because the SFS structure was defined up front by the WebMon user before classification began. Consequently, none of the documents retrieved during Web monitoring were considered suitable to classify into the other 24 folders. This highlights the weakness of defining a storage structure based on abstract domain knowledge. A user is restricted to defining the structure based on their own hierarchical understanding of the domain, and in most cases this understanding does not accurately reflect the real content of the domain itself.

Number of Nodes (concepts)	77
Total Nodes with Documents	76
Total Nodes without Documents	1
Number of Single Level Nodes	22
Average Child Nodes per Node	1.69
Average Attributes per Node	4.08

Table 4.2 – Summary of HCK Concept Lattice Structure

By comparing the physical composition of the SFS (see Table 4.1) with the HCK concept lattice structure (see Table 4.2), the implications of browsing documents based on heuristic classification knowledge as opposed to abstract domain knowledge can be made clear. The most obvious comparison that can be made in the current analysis is the difference between the average number of sub-folders per folder in the SFS, and the average number of child nodes per node in the HCK lattice. In the SFS there is an average of 6.08 sub-folders for every folder (excluding leaf folders), while in the HCK lattice there is an average of 1.69 children nodes per node. Since the SFS is a hierarchical tree structure, it would be traversed starting from the root folder and finishing at a leaf folder. This means that in browsing the SFS a user tries to pick the best sub-folder at each step in order to locate a particular document. Each time a document is not located in a particular folder, the user would have to make the decision between an average of about 6 sub-folders as to where to go next. This also means that if a leaf folder is reached, it is difficult to know what to do next because the best guesses have already been made at each decision point.

However, with the HCK lattice structure, making the decision of where to go next is much less overwhelming for the user. This is because on average there is only about 1 or 2 child nodes to choose from. Also, since the HCK lattice is more of a network type structure, it means that if a document is not located by taking one path, it is possible to go back up another path rather than starting again. This opens up new decisions which have not previously been considered.

A further interesting aspect of utilising the HCK lattice for browsing documents is that every node except one (which would be the bottom-most node) contains at least one document (see Table 4.2). However, in the SFS there are 24 folders that do not contain any documents (see Table 4.1). This means there are 24 possible decisions a user could make when browsing the SFS that are potentially useless in locating a particular document. This not only makes locating a document more difficult in the SFS, but it would no doubt also increase a user's frustration.

4.2.2. Comparison of Lattice Generated using Heuristic Classification Knowledge and Lattice Generated using a Combination of Knowledge Types

As an addition to the initial analysis, a second concept lattice was generated based on a combination of the MCRDR heuristic classification knowledge and abstract domain knowledge (HCK-ADK lattice). This analysis was conducted to determine the usefulness of utilising the terms from both types of knowledge for browsing. Table 4.3 shows the statistics gathered by analysing the physical composition of the HCK-ADK lattice.

Number of Nodes (concepts)	88
Total Nodes with Documents	87
Total Nodes without Documents	1
Number of Single Level Nodes	3
Average Child Nodes per Node	1.69
Average Attributes per Node	7.18

Table 4.3 – Summary of HCK-ADK Concept Lattice Structure

Comparing the physical structure of the HCK-ADK lattice (Table 4.3) with the structure of the HCK lattice (Table 4.2) produces some very interesting results. The most interesting result is the significant decrease in the amount of single level nodes

in the HCK-ADK lattice. In this analysis, a single level node is a node that has the supremum node (top most concept in the lattice) as its only predecessor, and the infimum node (bottom most concept in the lattice) as its only successor. If a large percentage of the total nodes in a lattice are single level nodes, it implies that the overall lattice structure is very shallow, meaning that more of the concepts will be general in nature. In regards to browsing the lattice for documents, this implies it will be more difficult for a user to locate the document desired. This is because there are fewer concepts in the lattice that would be specific enough to uniquely represent the attributes of that document.

Calculating the percentage of single level nodes in each lattice generated reveals that even though the HCK-ADK lattice contains 10 extra nodes (88 nodes) than the HCK lattice (77 nodes), only about 3 percent of nodes in the HCK-ADK lattice are single level nodes. However, in the HCK lattice, about 29 percent of all nodes are single level nodes. This implies that it would be much easier to locate a particular document when browsing the HCK-ADK lattice because a larger number of terms are being used to represent the attributes of documents. The result is that there are a greater number of more specific concepts in the which provides a much richer context for browsing.

4.3. Analysis of the Distribution of Documents

The second statistical analysis undertaken involved analysing how documents were distributed in the various browsing structures. The aim of this analysis was to determine whether utilising heuristic classification knowledge as a resource for browsing enhances a user's ability to locate a particular document.

4.3.1. Comparison of Storage Folder Structure and Lattice Generated using Heuristic Classification Knowledge

Table 4.4 shows the main statistics gathered from analysing the distribution of documents in the storage folder structure (SFS).

	Total	%
Documents in 1 Folder	1041	97.9
Documents in 2 Folders	17	1.6
Documents in 3 Folders	4	0.4
Documents in 4 Folders	1	0.1
Documents in 5 Folders	0	0.0
Documents in 6 Folders	0	0.0
Documents in 7 Folders	0	0.0
Documents in 8 Folders	0	0.0
Documents in more than 8 Folders	0	0.0

Table 4.4 – Distribution of Documents in Multiple Folders in Storage Folder Structure

The most significant result from analysing the distribution of documents in the SFS shows that the majority of the total 1063 classified documents are only located in a single folder. This implies that it would be quite difficult to locate a particular document when browsing the SFS because few documents can be found in multiple folders. Consequently, this makes the decision of which folders a user selects in searching for a document a lot more critical, since the likelihood of finding the document in a particular folder is relatively small.

The ability to locate a document can be significantly improved if the heuristic classification knowledge is used as a resource for browsing instead. This is obvious in the statistics that were gathered from analysing the distribution of documents in the concept lattice generated on MCRDR heuristic classification knowledge (HCK lattice). These results are shown in Table 4.5.

	Total	%
Documents at 1 Node	0	0.0
Documents at 2 Nodes	918	86.4
Documents at 3 Nodes	123	11.6
Documents at 4 Nodes	14	1.3
Documents at 5 Nodes	4	0.4
Documents at 6 Nodes	1	0.1
Documents at 7 Nodes	3	0.3
Documents at 8 Nodes	0	0
Documents at more than 8 Nodes	0	0

Table 4.5 – Distribution of Documents at Multiple Nodes in HCK Concept Lattice

In the HCK lattice, documents are distributed much more evenly than in the SFS. As a result, a larger amount of documents are located at a higher number of multiple locations (nodes) in the HCK lattice. This is also evident when the distribution of documents between the SFS and HCK lattice are compared graphically, as shown in Figure 4.1. Note that the document distribution for a concept lattice based on a combination of knowledge types (HCK-ADK concept lattice) is also shown.

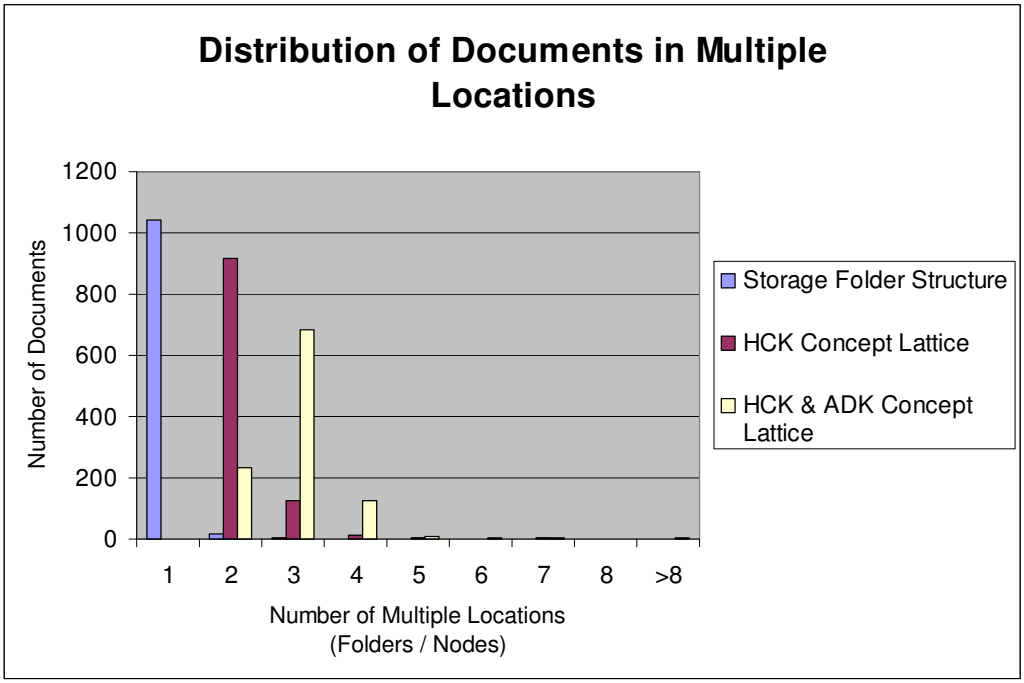

Figure 4.1 – Document Distribution for Storage Folder Structure and Concept Lattices

It is interesting to observe that some documents in the HCK lattice can even be found at up to 7 nodes, whereas in the SFS the largest number of multiple locations a document can be found at is in 4 folders. Furthermore, the majority of documents are found in 2 locations (nodes) in the HCK lattice, as opposed to in a single location in the SFS. From these results it can be concluded that browsing the HCK lattice provides far greater possibilities for locating a particular document.

4.3.2. Comparison of Lattice Generated using Heuristic Classification Knowledge and Lattice Generated using a Combination of Knowledge Types

As an addition to the initial analysis, statistics were also gathered on the distribution of documents in a concept lattice that was generated using the terms of both the

MCRDR heuristic classification knowledge and abstract domain knowledge (HCK-ADK lattice). Table 4.6 shows the results.

	Total	%
Documents at 1 Node	0	0.0
Documents at 2 Nodes	234	22.0
Documents at 3 Nodes	684	64.3
Documents at 4 Nodes	124	11.7
Documents at 5 Nodes	10	0.9
Documents at 6 Nodes	4	0.4
Documents at 7 Nodes	3	0.3
Documents at 8 Nodes	0	0.0
Documents at more than 8 Nodes	4	0.4

Table 4.6 – Distribution of Documents at Multiple Nodes in HCK-ADK Concept Lattice

It is interesting to note the effect that utilising the terms from both knowledge types has on the distribution of documents in the lattice structures. In the HCK-ADK lattice, the distribution of documents appears to be more evenly spread than in the HCK lattice. This can be clearly seen in Figure 4.1 (referred to previously). Also, in the HCK-ADK lattice, 78 percent of documents are located at 3 or more nodes, whereas only about 14 percent are located at that many nodes in the HCK lattice. This shows that the utilisation of the terms of both knowledge types can also provide more possibilities for locating a document while browsing.

4.4. Analysis of Browsing the Browsing Structures

The final statistical analysis undertaken involved simulating the way a user might actually browse each of the different structures. For the storage folder structure (SFS) this was simulated programmatically by beginning at the first level of browsing, namely the root folder, and recording information about the properties of that browsing level. Then the entire SFS was traversed one level (folder) deeper to all sub-folders visible from the first level, and the properties of that level were also recorded. This process continued until it was not possible to traverse any deeper, namely when all folders on the browsing level were leaf folders.

A similar programmatic simulation was also applied to the generated concept lattices to record the information about each level of browsing in the lattice structure. The deepest level of browsing in the lattice was the level that contained only the infimum

node (bottom most concept in the lattice). It should be noted that the structure of a concept lattice is such, that when browsing the lattice an individual node may appear (be visible) at two different browsing depths, depending on which path is taken through the lattice.

The statistics that were recorded at each level of browsing included the total number of folders or nodes for that level, the total number of documents, the total number of unique documents, and the average number of documents per folder or node on that level.

4.4.1. Comparison of Storage Folder Structure and Lattice Generated using Heuristic Classification Knowledge

Table 4.7 presents the statistics gathered by simulating browsing the storage folder structure (SFS). Table 4.8 presents those gathered by simulating browsing the concept lattice which was generated based on the MCRDR heuristic classification knowledge (HCK lattice).

Browsing Depth (folders)	Total Folders	Total Documents	Unique Documents	Average Documents per Folder
1 Level (root)	1	0	0	0.00
2 Levels	20	489	487	24.45
3 Levels	59	602	586	10.20

Table 4.7 – Analysis of Browsing the Storage Folder Structure

Browsing Depth (nodes)	Total Nodes	Total Documents	Unique Documents	Average Documents per Node
1 Level (root)	1	1063	1063	1063.00
2 Levels	46	1088	1063	23.65
3 Levels	25	152	145	6.08
4 Levels	6	9	7	1.50
5 Levels	1	2	2	2.00
6 Levels	1	0	0	0.00

Table 4.8 – Analysis of Browsing the HCK Concept Lattice

The first and perhaps most obvious comparison that can be made between browsing the two structures is the difference in the number of browsing levels. Starting at the root folder (level 1) in the SFS, it is possible to traverse to a maximum browsing

depth of 3 levels. On the other hand, in the HCK lattice it is possible to traverse to a maximum browsing depth of 6 levels. One might argue that since there are fewer levels of browsing in the SFS, it would be much easier for a user to browse. However, the fact that there are fewer levels of browsing means that the amount of folders on each level is quite large. The structure of the SFS is such, that the deeper the user browses, the larger the amount of folders that appear on each level. This means the decision of which folder to select when trying to locate a document becomes much more difficult with each new level that is traversed.

In the HCK lattice the opposite is the case. This can be clearly seen in Figure 4.2. Disregarding the first level of browsing (the root node), the deeper a user browses the HCK lattice structure, the fewer the nodes that appear at each browsing level. Therefore the decision of where to go next when browsing the HCK lattice only becomes easier rather than more difficult.

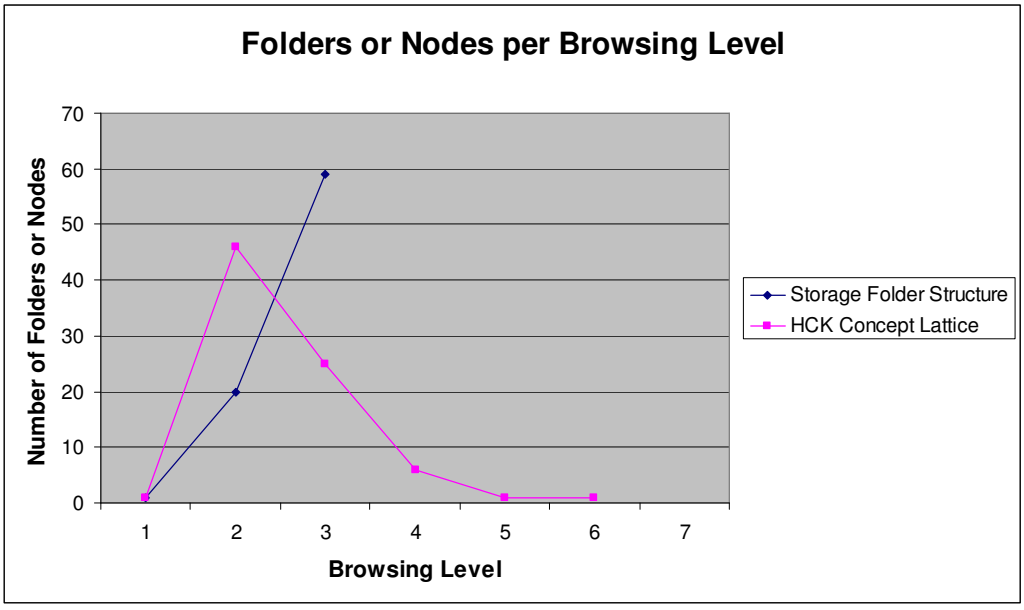

Figure 4.2 – Number of Folders or Nodes per Browsing Level

It is also interesting to compare the total number of documents and unique documents at each level of browsing in the SFS and the HCK lattice (see Table 4.7 and Table 4.8). Since the SFS only has three levels, it is appropriate to compare only the first three levels of both structures. This comparison reveals that all 1063 classified documents can be located at both of the first two levels of browsing in the

HCK lattice, while not even half of all the documents can be found at each of the same two levels of browsing in the SFS. This would suggest that there is more chance of locating a desired document in the HCK lattice as there is in the SFS.

However, one might argue that since there are just over twice as many nodes as folders on the second level of browsing in the HCK lattice, the chance of locating a document would not be easier than in the SFS because there are far more options a user has to choose from. Though this is the case, comparing the number of unique and total documents on the second level of browsing in each structure reveals that more documents are repeated on that level in the HCK lattice than in the SFS. This difference can also be easily recognised graphically, as shown in Figure 4.3. Note that the results from analysing a concept lattice generated on a combination of knowledge types (HCK-ADK concept lattice) are also shown.

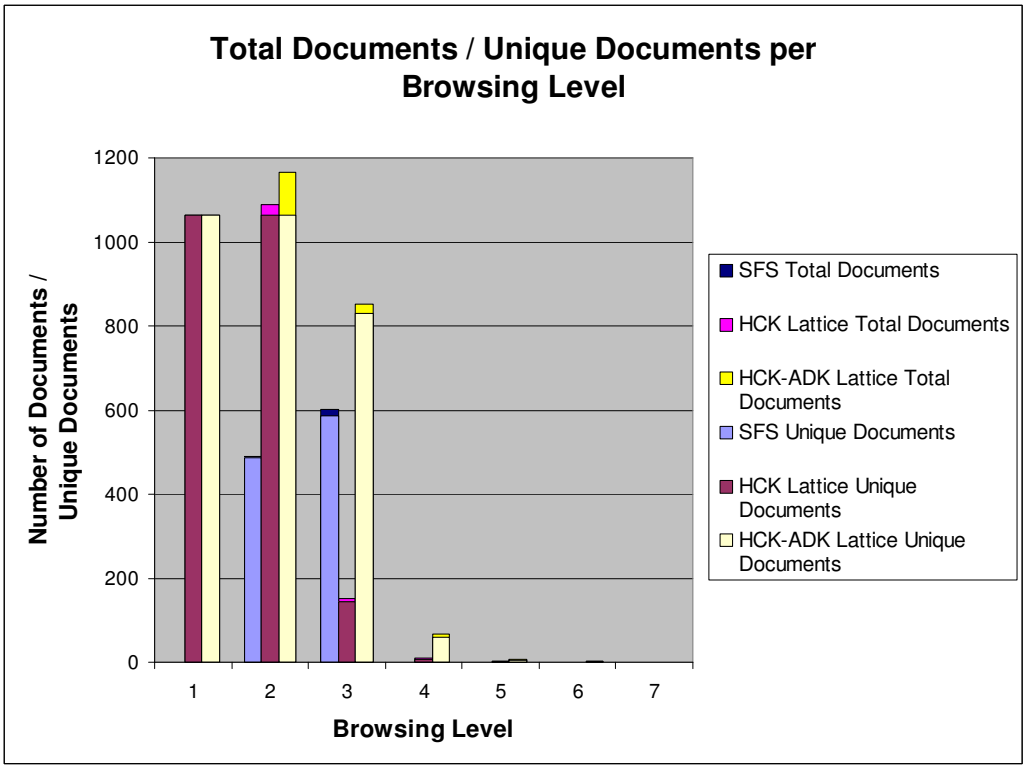

Figure 4.3 – Total Number of Documents per Browsing Level

Reviewing the data presented in Table 4.7 and Table 4.8 shows that in the SFS only 2 documents are repeated on the second level of browsing, whereas in the HCK lattice there are 25 documents that are repeated. It is also interesting to note that the

average number of documents per node on the second level of browsing in the HCK lattice is not much different to the average number of documents per folder on the same level in the SFS. Therefore, from the results presented, it can be concluded that even though there are more options a user has to select between on the second level of browsing in the HCK lattice, the chances of finding the correct document are still relatively high.

4.4.2. Comparison of Lattice Generated using Heuristic Classification Knowledge and Lattice Generated using a Combination of Knowledge Types

To determine the effect on browsing that utilising the terms from both types of knowledge would have, a second concept lattice was generated based on a combination of the MCRDR heuristic classification knowledge and abstract domain knowledge (HCK-ADK lattice). The statistics gathered from programmatically simulating the browsing of the HCK-ADK lattice are shown in Table 4.9.

Browsing Depth (nodes)	Total Nodes	Total Documents	Unique Documents	Average Documents per Node
1 Level (root)	1	1063	1063	1063.00
2 Levels	21	1166	1063	55.52
3 Levels	46	852	829	18.52
4 Levels	20	68	60	3.40
5 Levels	5	8	6	1.60
6 Levels	1	2	2	2.00
7 Levels	1	0	0	0.00

Table 4.9 – Analysis of Browsing the HCK-ADK Concept Lattice

Comparing the difference between the HCK-ADK lattice (Table 4.9) and the HCK lattice (Table 4.8) shows that there is only one extra level of browsing in the HCK-ADK lattice. Another interesting statistic is that the average number of documents per node on nearly all the levels of browsing in the HCK-ADK lattice is significantly higher than that in the HCK lattice. This difference is more obvious graphically, as shown in Figure 4.4.

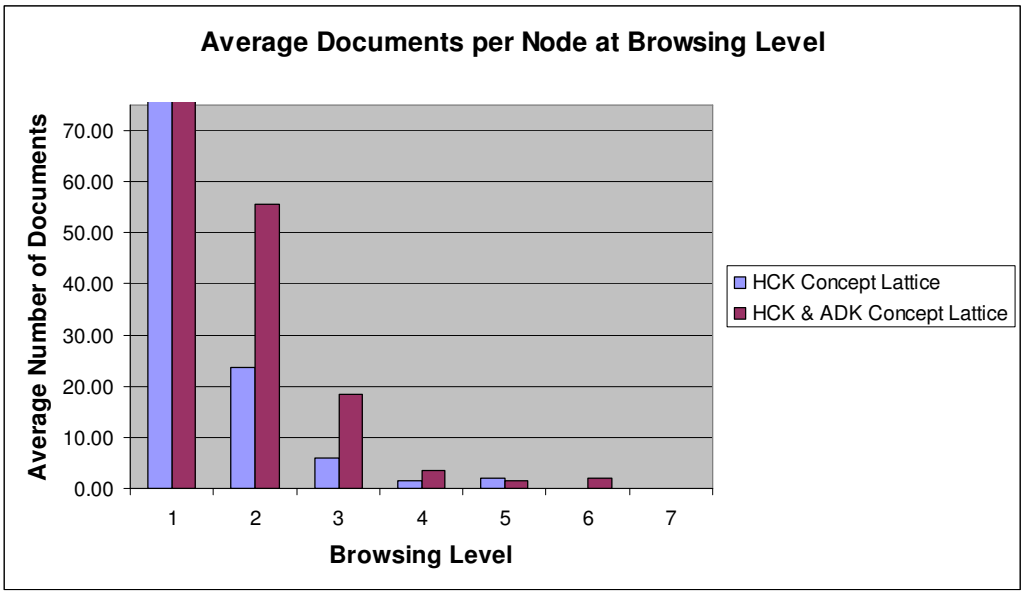

Figure 4.4 – Average Number of Documents per Node at Browsing Levels

Furthermore, as shown in Figure 4.3 (referred to previously), the overall difference between the number of total and unique documents on each level in the HDK-ADK lattice is also significantly higher than in the HCK lattice. Therefore, from the comparisons presented it can be concluded that the utilisation of the terms of both knowledge types improves the possibility of locating a document during browsing. This makes the browsing experience all the more beneficial for a user.

Chapter 5
Conclusion

5.1. Overview

The investigation undertaken in the study detailed in this thesis was aimed at determining the feasibility of utilising heuristic classification knowledge acquired through the use of MCRDR as a resource for browsing documents retrieved from the WWW. A Web-based system was developed which generated a FCA concept lattice which was constructed using the heuristic classification knowledge of MCRDR. To evaluate the feasibility of utilising heuristic classification knowledge as a resource for browsing documents, a comparative statistical analysis was performed. This involved comparing the difference between browsing documents using two different structures. Namely, a storage folder structure (SFS) based on abstract knowledge of a domain, and a concept lattice based on MCRDR heuristic classification knowledge.

From the evaluation performed, it is concluded that the concept lattice-based browsing scheme of FCA provides a feasible way to utilise MCRDR heuristic classification knowledge for browsing documents of a specific domain. An analysis of the physical composition of the SFS compared with the concept lattice structure revealed that browsing based on heuristic classification knowledge significantly simplifies each decision a user has to make during browsing. Also, analysing the distribution of documents in each browsing structure revealed that a user's ability to locate a particular document when browsing the lattice structure is significantly enhanced. Documents are more evenly distributed throughout the lattice than in the SFS, and they can also be found in a larger number of multiple locations. Furthermore, by programmatically simulating the way a user might browse each structure, it was possible to determine the options they would be presented with during browsing. Even though the lattice structure based on heuristic classification knowledge appeared to require more interaction from a user during browsing than when using the SFS, the browsing experience is much less overwhelming because each individual stage of browsing is much simpler.

In addition, the results of a secondary investigation concluded that using the terms of both abstract domain knowledge and heuristic classification knowledge also presents itself as a viable option for browsing documents. Statistically comparing a lattice generated on the terms of both knowledge types with a lattice generated plainly on heuristic classification knowledge produced some interesting results. The results showed that the utilisation of the terms of both knowledge types provides a much richer context for browsing. Each document can not only be found at a larger number of multiple locations in the lattice, but the extra terms also enable the location of each document to be identified more specifically.

5.2. Further Work

There are potentially several areas of research related to this study that can be investigated. An immediate continuation of the work undertaken might be to incorporate the prototyped concept lattice browsing approach of iWeb FCA into the iWeb Web Portal Site. This may be useful for providing an alternate method to users for browsing documents on that site, especially considering the significant quantity of information available.

An aspect that was not covered by this study is a user's actual satisfaction of browsing documents based on heuristic classification knowledge, as compared with browsing based on abstract domain knowledge. To evaluate this would also be interesting and would most likely involve performing a quantitative user study. The study could compare and assess the performance of browsing documents based on each type of knowledge.

It may also be interesting to investigate the use of other classification knowledge types as a resource for browsing documents. This study simply utilised the classification knowledge of MCRDR because it was readily available and suitable. There may well be other types of classification knowledge that can be utilised appropriately for browsing documents. In the same manner, it may also be useful to evaluate the use of an alternate browsing structure, other than the concept lattice of

FCA, that can also utilise heuristic classification knowledge as a resource for browsing documents.

However, perhaps the most interesting point that remains to be seen is whether browsing schemes based on heuristic classification knowledge will become a standard for browsing information on the WWW. With the consistent increase in the amount of information being generated on the WWW, there is an increasing need for more effective and simple ways of locating and retrieving information. To this extent, the utilisation of heuristic classification knowledge as a resource for browsing and searching of information may provide a potential solution to this problem.

References

Boyapati, V., Chevrier, K., Finkel, A., Glance, N., Pierce, T., Stockton, R. and Whitmer, C. 2002, 'ChangeDetector: a site-level monitoring tool for the WWW', *Eleventh International Conference on World Wide Web*, Honolulu, Hawaii, USA

Buchwitz, L. 1997, 'Monitoring Competitive Intelligence using Internet Push Technology', *Competitive Intelligence Review*.

Chakravarthy, S., Sanka, A., Jacob, J. and Pandrangi, N. 2004, 'A Learning-Based Approach for Fetching Pages in WebVigiL', *2004 ACM Symposium on Applied Computing*, ACM, Nicosia, Cyprus, pp. 1725-1729

Chin, P. 2003, *Push Technology: Still Relevant After All These Years?*, Intranet Journal, viewed October 8, <http://www.intranetjournal.com/articles/200307/ij_07_23_03a.html>

Cho, W. C. 2003, 'Use of Cache Mechanism for Web Information Search', Masters thesis, University of Tasmania.

Compton, P. and Jansen, R. 1989, 'A Philosophical Basis for Knowledge Acquisition', *3rd European Knowledge Acquisition for Knowledge-Based Systems Workshop*, Paris, pp. 75-89

Correia, J. H., Stumme, G., Wille, R. and Wille, U. 2003, 'Conceptual Knowledge Discovery - A Human-Centred Approach', *Applied Artificial Intelligence*, vol. 17, pp. 281-302.

Dazely, R. and Kang, B. H. 2003, 'Weighted MCRDR: Deriving Information about Relationships between Classifications in MCRDR', *Australian Conference on Artificial Intelligence*, pp. 245-255

Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R. and Robbins, D. C. 2003, 'Stuff I've Seen: A System for Personal Information Retrieval and Re-use', *26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Toronto, Canada, pp. 72-79

Edwards, G., Compton, P., Malor, R., Srinivasan, A. and Lazarus, L. 1993, 'PEIRS: a pathologist maintained expert system for the interpretation of chemical pathology reports', *Pathology*, vol. 25, pp. 27-34.

Furnas, G. W. 1986, 'Generalised Fisheye Views', *Proceedings of CIII'86*, ACM, pp. 16-23

Ganter, B. and Wille, R. 1997, 'Applied Lattice Theory: Formal Concept Analysis', *Preprints, http://wwwbib.mathematik.tu-darmstadt.de/Math-Net/Preprints/Listen/pp97.html*.

Glance, N., Meunier, J.-L., Bernard, P. and Arregui, D. 2001, 'Collaboritive Document Monitoring', *2001 International ACM SIGGROUP Conference on Supporting Group Work*, ACM, Boulder, Colorado, USA

Godin, R., Missaoui, R. and Alaoui, H. 1995, 'Incremental concept formulation algorithms based on Galois (concept) lattices', *Computational Intelligence*, vol. 11, no. 2, pp. 246-267.

Godin, R., Pichet, C. and Gecsei, J. 1989, 'Design of a browsing interface for information retrieval', *12th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, pp. 32--39

Greenspan, R. 2002, *Search Engine Usage Ranks High*, viewed 9 June, <www.clickz.com/stats/markets/advertising/article.php/5941_1500821>

InternetWorldStats 2004, *Internet Usage Statistics - The Big Picture: World Internet Users and Population Stats*, viewed October 8, 2004, <http://www.internetworldstats.com/>

Kang, B. H. 1996, 'Multiple Classification Ripple Down Rules', PhD thesis, University of New South Wales.

Kang, B. H., Compton, P. and Preston, P. 1995, 'Multiple Classification Ripple Down Rules: Evaluation and Possibilities', *9th Banff Knowledge Aquisition for Knowledge Based Systems Workshop*, Banff, pp. 17.11-17.20

Kang, B. H., Yoshida, K., Motoda, H. and Compton, P. 1997, 'Help Desk System with Intelligent Interface', *Applied Artificial Intelligence*, vol. 11, pp. 611-631.

Kim, M. 2003, 'Document Management and Retrieval for Specialised Domains: An Evolutionary User-Based Approach', Ph.D. thesis, University of New South Wales.

Kim, M. and Compton, P. 2000, 'Developing a domain-specific Document Retrieval Mechanism', *6th Pacific Knowledge Acquisition Workshop*, Sydney, Australia, pp. 189-206

Kim, M. and Compton, P. 2001, 'A Web-based Browsing Mechanism Based on Conceptual Structures', *9th International Conference on Conceptual Structures*, Standford University, California, USA, pp. 47-60

Kim, M. and Compton, P. 2004, 'Evolutionary document management and retrieval for specialized domains on the web', *International Journal of Human-Computer Studies*, vol. 60, no. 2, p. 201–241.

Kim, Y. S., Park, S. S., Deards, E. and Kang, B. H. 2004a, 'Adaptive Web Document Classification with MCRDR', *International Conference on Information Technology (ITCC)*, Las Vegas, NV, USA

Kim, Y. S., Park, S. S., Kang, B. H. and Choi, Y. J. 2004b, 'Incremental Knowledge Management of Web Community Groups on Web Portals', *5th International Conference on Practical Aspects of Knowledge Management*, Vienna, Austria

Kobayashi, M. and Takeda, K. 2000, 'Information Retrieval on the Web', *ACM Computing Surveys*, vol. 32, no. 2, pp. 144-173.

Lam, S. K. S. and Ozsu, M. T. 2002, 'Querying Web data - the WebQA approach', *Third International Conference on Web Information Systems Engineering*, IEEE, Singapore, pp. 139-148

Mladenic, D. 1999, 'Text-learning and related intelligent agents', *Application of Intelligent Information Retrieval*

Park, S. S., Kim, Y. S. and Kang, B. H. 2003, 'Web Information Management System: Personalization and Generalization', *IADIS International Conference WWW/Internet 2003*, Algarve Portugal

Preston, P., Compton, P., Edwards, G. and Kang, B. H. 1996, 'An Implementation of Multiple Classification Ripple Down Rules', *Tenth Knowledge Acquistion for Knowledge-Based Systems Workshop*.

Rajapakse, R. K. and Denham, M. 2003, 'A Reinforcement Learning Strategy for (formal) Concept and Keyword Weight Learning for Adaptive Information Retrieval', *9th International Conference on User Modeling*, Johnstown, Pennsylvania, USA, pp. 29-39

Richards, D. C. 1998, 'The Reuse in Ripple Down Rule Knowledge Based Systems', Doctorate thesis, University of New South Wales.

Richards, D. C. 2001, 'Combining Cases and Rules to Provide Contexualised Knowledge Based Systems', *Third International Conference on Modelling and Using Context*, Coff's Harbour, Australia, pp. 85-94

Richards, D. C. and Compton, P. 1997, 'Combining Formal Concept Analysis and Ripple Down Rules to Support the Reuse of Knowledge', *Ninth International Conference on Software Engineering Knowledge Engineering SEKE'97*, Springer Verlag, Madrid, Spain

Senastiani, F. 2002, 'Machine Learning in Automated Text Categorization', *ACM Computing Surveys*, vol. 34, no. 1, pp. 1-47.

Tam, G. K. T. 2004, 'FOCAS - Formal Concept Analysis and Text Similarity', Honours thesis, Monash University.

Wille, R. 1982, 'Restructuring lattice theory: an approach based on hierarchies of concepts', in I. Rival (ed.), *Ordered Sets*, Reidel, Dordrecht-Boston**:** pp. 445-470.

Wille, R. 1989, 'Lattices in Data Analysis: How to Draw them with a Computer', in I. Rival (ed.), *Algorithms and Order*, Kluwer, Dordrecht, Boston**:** pp. 33-58.

Wille, R. and Ganter, B. 1999, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin-Heidelberg.

Appendix A
Listing of Software Available on CD

A.1 Software Used in Development

(/dev_software/***apache_2.0.52-win32-x86-no_ssl.msi***) - Apache Web Server

(/dev_software/***php-4.3.9-Win32.zip***) - PHP Module

(/dev_software/***mysql-4.0.22-win.zip***) - MySQL API

(/dev_software/***phpMyAdmin-2.6.0-pl2.zip***) - MySQL Admin

A.2 Software Developed or Modified

A.2.1 iWeb FCA (Setup Files)

(/iweb_fca/conf/***config.php***) - Main Configuration File

(/iweb_fca/inc/v09/***func_db.php***) - MySQL Database Settings

(/iweb_fca/scripts/table_creation/***iwebfca_tables.sql***) - iWeb FCA Database Table Creation Script

(/iweb_fca/scripts/input_data/***ehealth/***) - Concept Lattice Experimentation Data

(/iweb_fca/***index.php***) - iWeb FCA Main Menu Page

A.2.2 iWeb Web Portal Site (Setup Files)

(/iweb_portal/conf/***config.php***) - Main Configuration File

(/iweb_portal/inc/v09/***func_db.php***) - MySQL Database Settings

(/iweb_portal/scripts/***iWeb_eHealth_Tables_Data.zip***) - iWeb Portal Database Table Creation Script

(/iweb_portal/***_SETUP_README.txt***) - Readme Explaining General Setup

(/iweb_portal/***index.php***) - Main Portal Page